Goal recognition from incomplete action sequences by probabilistic grammars

Ryosuke Kojima and Taisuke Sato

Tokyo Institute of Technology

Abstract. This paper summarizes recent progress concerning prefix probability computation for PCFGs in a logic-based modeling language PRISM. A prefix is an initial substring of a sentence and the prefix probability computation is already introduced to PRISM but applications are still scarce. We report a new application to web data that identifies visitors' intentions, or goals visiting a website from their action sequences using prefix probability.

Keywords: PRISM PCFG "prefix probability" "goal recognition"

1 Introduction

Probabilities are computed in a logic-based modeling language PRISM via explanation graphs using dynamic programming [1]. When probability computation requires an infinite sum of probabilities, typically for probabilistic model checking using Markov chains and prefix probability computation in PCFGs, it is still possible to compute probabilities using dynamic programming while solving a system of linear probability equations.

We here report[1] an application of our previous work [2] which introduced to PRISM a basic mechanism of prefix probability computation by cyclic explanation graphs. We apply prefix probability computation to the problem of estimating the goals of visitors who visit a website from their session log data. In our setting, visitors are a mixture of various groups with different goals. We analyze their action sequence as a sentence in a PCFG specialized for each group or goal, and model the whole session log data as a mixture of PCFGs, i.e., a combination of such specialized PCFGs. Then the task is to estimate the visitor's goal as a most likely start symbol in the mixture of PCFGs.

This approach has two problems however. One is that we cannot obtain any information until visitors complete their actions as sentences in a PCFG. So it is impossible to obtain visitor information online and guide them to a target page, say, by displaying affiliate links. The other one is the problem of *unachieved visitor*s. *Unachieved* visitors are those who quit the site for some reason before they achieve their goals or purposes. Obviously their action sequences should be considered as incomplete, not as complete sentences. We solve these problems

[1] This paper summarizes part of the paper "Prefix and infix probability computation in PRISM" to be presented at PLP 2014 workshop.

by generalizing sentences to prefixes and by considering action sequences as prefixes in a PCFG. A prefix is an initial substring of a sentence and since any incomplete sequence by an unachieved visitor can be considered as a prefix, this generalization makes it possible to identify a visitor's goal online and estimate the most likely parse tree for it.

2 Prefix probability computation

A PCFG $\mathsf{G}_{\mathbf{\Theta}}$ is a CFG G augmented with a parameter set $\mathbf{\Theta} = \bigcup_{N^i \in \mathbf{N}} \{\theta_r\}_{N^i}$ where $\mathbf{N}$ is a set of nonterminals, and N^1 a start symbol and $\{\theta_r\}_{N^i}$ the set of parameters associated with rules $\{r \mid r = N^i \to \zeta\}$ for a nonterminal N^i where ζ is a sequence of nonterminal and terminal symbols . We assume that the θ_r's satisfy $0 < \theta_r < 1$ and $\sum_{\zeta: N^i \to \zeta} \theta_{N^i \to \zeta} = 1$.

A *prefix* $\mathbf{v}$ is an initial substring of a sentence and the prefix probability $P^{N^1}_{\mathrm{pre}}(\mathbf{v})$ of $\mathbf{v}$ is defined as an infinite sum of probabilities of sentences extending $\mathbf{v}$:

$$P^{N^1}_{\mathrm{pre}}(\mathbf{v}) = \sum_{\mathbf{w}} P_{\mathsf{G}}(\mathbf{vw})$$

where $\mathbf{w}$ ranges over strings such that $\mathbf{vw}$ is a sentence in G. There are a couple of methods to compute prefix probabilities in PCFGs [3] but they are computed in PRISM by way of cyclic explanation graphs generated by parsing using a prefix parser for PCFGs [2].

Let $\mathsf{G}_0 = \{\ \mathtt{s} \to \mathtt{s}\,\mathtt{s} : 0.4,\ \mathtt{s} \to \mathtt{a} : 0.3,\ \mathtt{s} \to \mathtt{b} : 0.3\ \}$ be a PCFG where "$\mathtt{s}$" is a start symbol and "$\mathtt{a}$" and "$\mathtt{b}$" are terminals and consider the prefix probability $P^{\mathtt{s}}_{\mathrm{pre}}(\mathtt{a})$ of prefix $\mathtt{a}$. To compute it we first parse "$\mathtt{a}$" as a prefix by the PRISM program DB_0 below.

```
values(s,[[s,s],[a],[b]]).
:- set_sw(s,[0.4,0.3,0.3]).

pre_pcfg(L):- pre_pcfg([s],L,[]).          % (1) L is a prefix
pre_pcfg([A|R],L0,L2):-                    % (2) L0 is ground when called
   ( get_values(A,_) -> msw(A,RHS),        % (3) if A is a nonterminal
       pre_pcfg(RHS,L0,L1)                 % (4) select rule A->RHS
   ;  L0=[A|L1] ),                         % (5) else consume A in L0
   ( L1=[] -> L2=[]                        % (6) (pseudo) success
   ; pre_pcfg(R,L1,L2) ).                  % (7) recursion
pre_pcfg([],L1,L1).                        % (8) termination
```

Fig. 1. Prefix parser DB_0

DB_0 is a prefix parser for PCFG G_0 which is encoded by the `values` declaration and `set_sw` command. As can be seen from the comments, it runs exactly like a standard top-down CFG parser except *pseudo success* at line `(6)`. *pseudo success* means an immediate return with success on the consumption of the input prefix `L1` ignoring the remaining nonterminals in `R` at line `(2)`[2].

By running a built-in command `?-probf(pre_pcfg([a]))`, we obtain a cyclic explanation graph for `a` which is a set of boolean formulas defining intermediate goals by equivalence formulas. It represents all possible SLD derivation paths of the top-goal `pre_pcfg([a])` in terms of AND/OR formulas comprised of `msw` atoms representing probabilistic choices. Then we convert the equivalence formulas to a set of probability equations. In general, a set of probability equations generated from prefix parsing by DB_0 is always linear. Solving the set of equations by matrix operation gives the prefix probability $P^{\mathsf{s}}_{\mathrm{pre}}(\mathtt{a})$ for `a` [2].

3 Action sequences as prefixes in a PCFG

Here we tackle the problem of identifying the purposes, or goals of visitors who visit a website from their session logs. We first abstract a visitor's session log into a sequence of actions comprised of five basic actions: `up`, `down`, `sibling`, `reload` and `move`. The first two, `up` and `down`, describe that a visitor moves respectively to a page in the parent directory and a subdirectory in the site's directory structure. An action `sibling` says that a visitor moves to a page in a subdirectory of the parent directory. An action `reload` means that a visitor requests the same page. An action `move` categorizes remaining miscellaneous actions. Moving between web pages is expressed by a sequence of basic actions. For example moving from /top/index.html to /top/child/a.html is a `down` action and moving from /top/index.html to /top/b.html is a `sibling` action.

Table 1. CFG rules

S → Survey
Survey → Search Destination
Search → Down Up Search | Down
Destination → `sibling` Destination | `sibling`
Down → Down `down` | `down`
Up → Up `up` | `up`

We consider an action sequence generated by a visitor who has achieved the intended goal as a sentence in a PCFG. We parse it using rules like those in Table 1 and obtain a parse tree. The CFG rules in Table 1 describe the visitors'

[2] This is justified because we assume the consistency of PCFGs that implies the probability of remaining nonterminals in `R` yielding some terminal sequences is 1.

goal-subgoal structure behind the action sequence. For example, the second and fourth rules say that when visitors' intention is Survey, they perform Search for a target page, reach a Destination page and look around the Destination page.

Since various visitors visit a website with different goals, we capture action sequences $\mathbf{w}$ made by visitors in terms of a mixture of PCFGs $P(\mathbf{w} \mid N^1) = \sum_A P^A(\mathbf{w} \mid A)P(A \mid N^1)$ where $P^A(\mathbf{w} \mid A)$ is the probability of $\mathbf{w}$ being generated by a visitor whose goal is represented by a nonterminal A and $P(A \mid N^1)$ is the probability of A being derived from the start symbol N^1 respectively. We call such A *goal-nonterminal* and assume that there is a unique rule $N^1 \to A$ for each goal-nonterminal A and also assume that it has a parameter $\theta_{N^1 \to A} = P(A \mid N^1)$.

Finally to estimate visitor goals online and also from action sequences generated by unachieved visitors, we replace a sentence probability $P^A(\mathbf{w} \mid A)$ in a mixture of PCFGs $P(\mathbf{w} \mid N^1) = \sum_A P^A(\mathbf{w} \mid A)P(A \mid N^1)$ by a prefix probability $P^A_{\mathrm{pre}}(\mathbf{w} \mid A)$. Hence given a prefix $\mathbf{w}_k$ with length k as an action sequence, we estimate the most likely goal-nonterminal $A^\dagger$ for $\mathbf{w}_k$ by

$$A^\dagger = \operatorname*{argmax}_A P^A_{\mathrm{pre}}(\mathbf{w}_k)\theta_{N^1 \to A} \tag{1}$$

where A ranges over possible goal-nonterminals.

4 Comparative experiment

In this section, we empirically evaluate our method, the *prefix method*, which is represented by (1). We compare it to two other methods: the PCFG method and logistic regression. The PCFG method naively uses a PCFG. It applies a mixture of PCFGs to action sequences $\mathbf{w}_k$. So every sequence is considered as a sentence and the most likely goal-nonterminal A^* is estimated from $\mathbf{w}_k$ by

$$A^* = \operatorname*{argmax}_A P^A(\mathbf{w}_k)\theta_{N^1 \to A} \tag{2}$$

where A ranges over possible goal-nonterminals. $P^A(\mathbf{w}_k)$ is the probability of $\mathbf{w}_k$ being derived by a component PCFG for goal-nonterminal A.

We also compare the prefix method with logistic regression which is a standard discriminative model that does not assume any structure behind data like the prefix and PCFG methods. For a fixed length k, the most likely visitor goal is estimated from $\mathbf{w}_k$ considered as a feature vector where features are five basic visitor actions introduced in Section 3.

We prepared three data sets of action sequences by preprocessing web server logs of U of S (University of Saskatchewan), ClarkNet and NASA in the Internet Traffic Archive, each referred to here as U of S, ClarkNet and NASA, containing 652, 4523 and 2014 action sequences respectively (details omitted). We also prepared a PCFG, *universal session grammar*, that has five goal-nonterminals,

Table 2. Result of clustering

Goal-nonterminal	Features and major action
Survey	`up`/`down` moves in the hierarchy of a website
News	`up`/`down` moves in the hierarchy of a website + reload a same page
Survey(SpecificAreas)	access to the same layer
News(SpecificAreas)	access to the same layer + reload a same page
Other	other actions

102 rules and 32 nonterminals. The meaning of the five goal-nonterminals is described in Table 2.

Visitors belonging to the goal nonterminal Survey survey wide area of a website and those belonging to News search for news and updates of a website. Two goal nonterminals Survey(SpecificAreas) and News(SpecificAreas) are similar to these but contain visitors who tend to stay in a specific area of a website. Other visitors including those who don't move in a web site along directory structure belong to Other.

We applied the above mentioned three methods to the task of estimating visitors' goals from prefixes of action sequences and compared their accuracy while varying prefix length. We also added a mixture of hidden Markov models (HMMs) in the comparison as a reference method.

To prepare a teacher data set to measure accuracy, we need to label each action sequence of a visitor by the visitor's true intention or goal, which is impossible. As substitution, we defined a *correct top-goal* for an action sequence in a data set to be the most likely goal-terminal for the sequence estimated by a mixture of PCFGs with the universal session grammar whose parameters are learned by MLE from the data set. This strategy seems to make sense as long as the universal session grammar is reasonably constructed.

In the experiment, accuracy was measured by five-fold cross-validation for each prefix length k ($2 \leq k \leq 20$). After parameter learning by a training data set, prefixes with length k were cut out from action sequences in the test set and their most likely goal-nonterminals were compared against correct top-goals labeling them. Fig. 2 shows accuracy for each k with standard deviation in which Prefix denotes the prefix method, PCFG the PCFG method[3] and Log-Reg logistic regression respectively. We also added HMM for comparison which uses a mixture of HMMs instead of a mixture of PCFGs.

Fig. 2 clearly demonstrates that the prefix method and the PCFG method outperform logistic regression and HMM, a standard discriminative model and a standard generative model without grammatical information respectively, when

[3] We applied a PCFG to prefixes by regarding them as sentences. In this experiment, the universal session grammar fails to parse at most two sequences for each dataset, so we ignore these sequences.

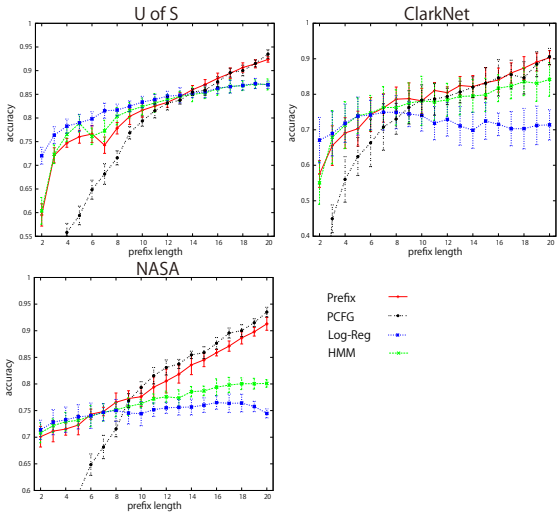

Fig. 2. Accuracy for U of S, ClarkNet and NASA

prefix is long. We note that all differences at prefix length $k = 20$ in the graph are statistically significant and confirmed by t-test at 0.05 significance level. Also we can observe that the PCFG method rapidly deteriorates when prefix gets shorter though the prefix method keeps fairly good performance comparable to logistic regression and HMM.

5 Conclusion

We have presented an application of prefix probability computation for PCFGs in PRISM to goal recognition of visitors from their action sequences who visit a website. A comparative experiment using three real datasets is conducted including the prefix method we proposed. The result demonstrates the superiority of the prefix method and the mixture of PCFGs for long action sequences over logistic regression and a mixture of HMMs.

References

1. Sato, T., Kameya, Y.: Parameter learning of logic programs for symbolic-statistical modeling. Journal of Artificial Intelligence Research **15** (2001) 391–454
2. Sato,T. and Meyer,P.: Infinite probability computation by cyclic explanation graphs. Theory and Practice of Logic Programming (TPLP) DOI: `http://dx.doi.org/10.1017/S1471068413000562`, published online, Nov. 04 (2013) 1–29
3. Jelinek, F., Lafferty, J.: Computation of the probability of initial substring generation by stochastic context-free grammars. Computational Linguistics **17**(3) (1991) 315–323